

Inxhinieria Softuerike

Modelet e procesit për zhvillimin e softuerit
(linear, iterative, agile)

Ramiz HOXHA
ramiz.hoxha@ubt-uni.net
2020/2021

Projekti i Softuerit

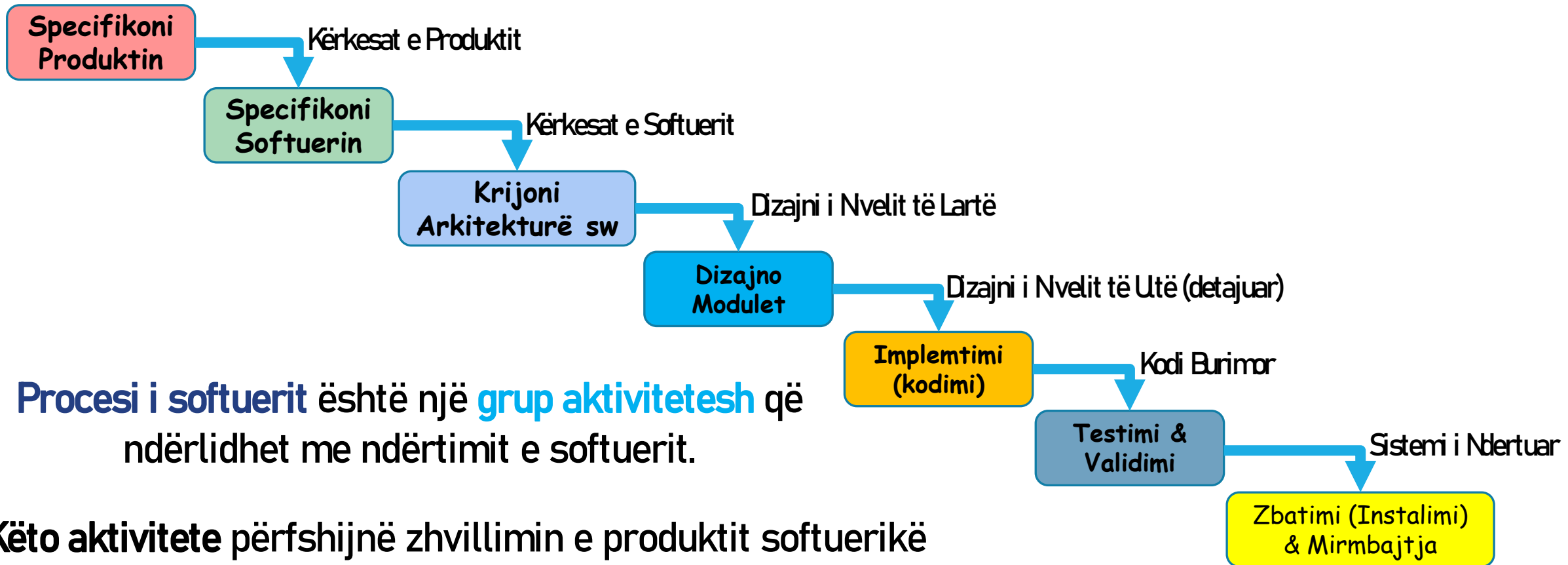
- Një **projekt** është një **grup i veprimeve** të *planifikuara* dhe të *realizuara* nga një **grup njerëzish** për të *arritur një qëllim të caktuar*, duke u kufizuar nga një **afat-kohor**, dhe ka një **kosto të caktuar**.

- Për të menaxhuar një projekt është e nevojshme të:
 - Përcaktoni metodologjinë e zhvillimi të softuerit.
 - Menaxhoni njerëzit e përfshirë
 - Menaxhoni kufizimet teknike
 - Menaxhoni mjetet e disponueshme ...

Procesi i Zhvillimit të Softuerit

- ❑ Metodologjia e Zhvillimit të Softuerit - është **korniza e përdorur** për të **planifikuar, kontrolluar dhe strukturuar** procesin e zhvillimit të sistemit softuerik.
- ❑ Gjithashtu quhet:
 - **Cikli i jetës për zhvillimin e softuerit**
 - **Procesi i zhvillimit të softuerit**
- ❑ Përfshinë **aktivitete**, qëllimi i të cilave është **zhvillimi** ose **përmisim** i softuerit.
- ❑ **Aktivitetet** e përgjithshme në të gjitha proceset e softuerit
 - **Specifikimi/Analiza** (Definimi i kërkesave të sistemit dhe përdoruesit).
 - **Dizajnimi** (Dizajnimi i arkitektures, programit, GUI, etj).
 - **Zhvillimi** (Kodimimi i sistemit)
 - **Validimi** (Testimit dhe Miratimi)
 - **Evuluimi** (Mirmbajtja dhe Zbatimit/Dorzimi)

Procesi i Zhvillimit të Softueri...



Procesi i softuerit është një **grup aktivitesh** që ndërlidhet me ndërtimit e softuerit.

Këto aktivitete përfshijnë zhvillimin e produktit softuerikë nga fillimi e deri në përfundim.

Procesi i Zhvillimit të Softueri...

- Megjithatë, një **process softuerik** përfshin gjithashtu përshkrimin e procesit i cili përfshin:
 - **Produktet**: *rezultatet* e një aktiviteti. p.sh, **Specifikimi i Kërkesës për Softuerin (SRS)** ndoshta një **model** për **arkitekturën e softuerit**.
 - **Rolet**: *përgjegjësitë* e njerëzve të përfshirë në proces. p.sh, **menaxheri i projektit, programuesi**, etj.
 - **Kushtet para** dhe **pas**: kushtet që duhet të jenë të *përfshira* para dhe pas një aktiviteti.
 - p.sh, *kushtet paraprake* të **dizajnit arkitekturor** janë **kërkesat e miratuara nga klienti**, ndërsa gjendja pas është **diagramet që përshkruajnë arkitekturën** që janë shqyrtuar.

Procesi i Zhvillimit të Softueri...

□ **Cikli i jetesës së softuerit:**

- Përcakton të gjitha **fazat** e zhvillimit të softuerit, duke filluar nga **specifikimet e kërkesave** të klientit, deri në **fazat** e fundit të **krijimi i softuerit**.

□ Këto **faza** janë të **organizuara** sipas **modeleve** që **drejtojnë (guide)** inxhinierin në aktivitetet e tij.

- **Modelet Lineare** (Linear models)
- **Modelet Përsëritëse** (Iterative models)
- **Modelet Rritëse** (Incremental models)
- **Modelet Adaptive** (Adaptive models)
- **Modelet Ekstreme** (Extreme models)

Modelet e zhvillimit të Softuerit

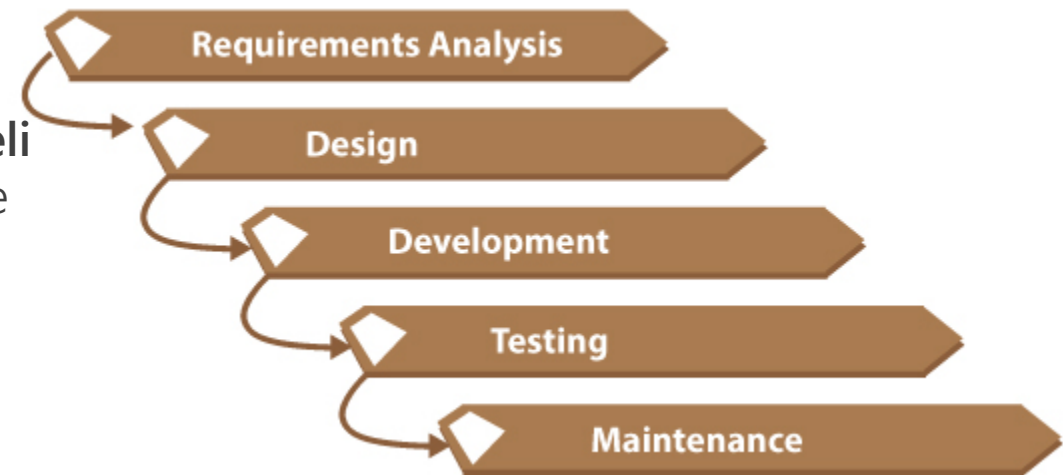
Modelet e zhvillimit të Softuerit

□ Modeli Linear

- **Aktivitetet ose fazat e varura** që janë ekzekutuar në një mënyrë **sekuenciale** duke mos ofruar ciklin e **regimit** (feedback).
- **Zgjidhja** ofrohet vetëm në **fazën finale** (përfundimtare).
- **Qëllimet, kërkesat dhe zgjidhjet e definuara mirë, ofrojnë shumë-pak ndryshim për kërkesat e fushëverpimit të projektit.**

- **Modeli linear-sekuencial** referohet gjithashtu si **Modeli Ujëvara** i cili është shumë i thjeshtë për tu **kuptuar** dhe **përdorur**.
- Në këtë model (**modeli ujëvarës**) **secila fazë duhet të përfundojë** para se të fillojë **faza tjetër** dhe nuk ka **mbivendosje** (overlapping) të fazave.

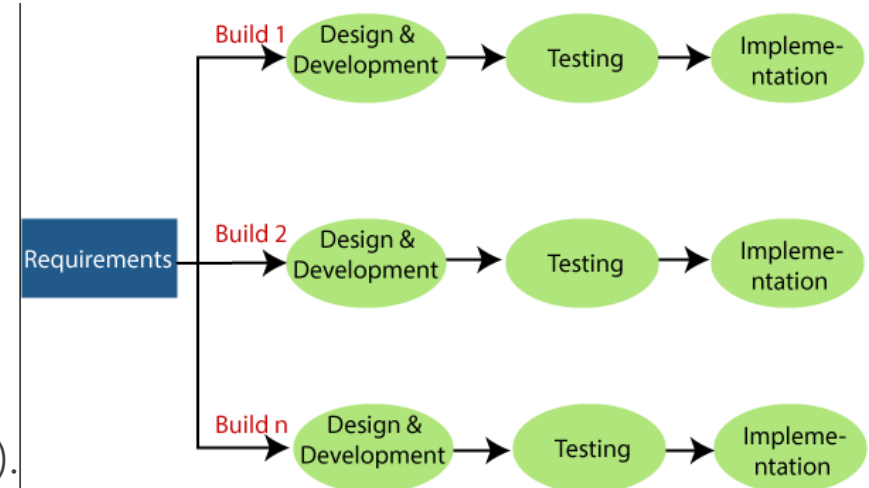
Modeli Linear-Sekuencial



Modelet e zhvillimit të Softuerit

□ Modeli Rritës (Incremental)

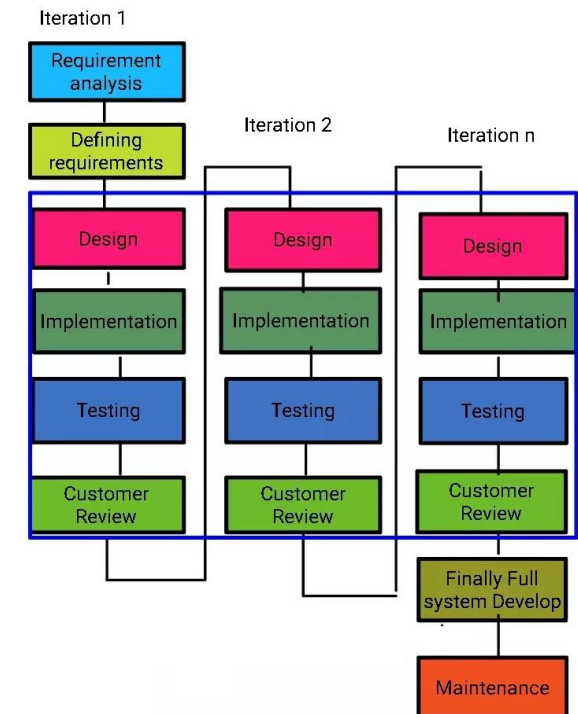
- Njësot si modeli linear, përveç se për secilen **fazë (iteracion)** lëshohet (ofrohet) një zgjidhje e pjesshme ose **deliverable**.
- Këto **iteracione** janë zakonisht **komplimente** për njëra-tjetrën. Pasi të kemi përfunduar të gjitha **iteracionet**, **produkti** është i **plotë**.
- Qëllimet, kërkesat dhe zhvillimi është bërë korrekt pa pasur nevojë të e kthehemi mbrapa, (orar më agresiv.)
- **Qasje rritëse** me e dobishme në krahasim me atë **sekuenciale**, veçanërisht kur madhësia e projektit nuk është e vogël.
- **Minimizon rrezikun** pasi secili inkrement (iteracion) mund të ofrohet në 2-3 muaj.
- Ciklet e **shkurta** të rritjes *përmirësojnë bashkëpunimin e klientit* me atë të *zhvillimit* (zhvilluesit).
- Ofron **produktin final** në formë të **komponenteve** (builds).



Modelet e zhvillimit të Softuerit...

□ Modeli Përsëritës (Iterative)

- Fazat përsëriten, kemi **mundesi feedback'ut** pas përfundimit të secilit **iteracion**.
- Në fund të iteracionit mund të **rezultoj me një zgjidhje të pjesshme** ose **përfundimtare**.
- **Qëllimet e definuara mirë, jo të gjitha funksionet (karakteristikat) e njohura, strategjia learn-by-doing**
- **Aftësia për t'i përfshirë ndryshimet në çdo iteracion**
- **Iteracionet e vogëla ofrojnë *bashkëpunim* me të shpeshtë me klientin** në fund të **çdo** iteracioni.
- Ofron **verzione** dhe **feedback** më **të hershëm** mbi produktin, **zbut rrezikun** e hendekut në **pritjet e klientit** dhe të kuptuarit të ekipit të zhvilluesve.
- **Feedback** (Reagimet) përdoren për të **rishikuar prioritetet** e projektit dhe për të **bërë ndryshime** në kërkesat, funksionalitetet, planifikimet,
Një **qasje iterative** më e dobishme krahasuar me **qasjen rritëse**, posaçërisht kur kërkesat e projektit pritet të ndryshojnë shumë

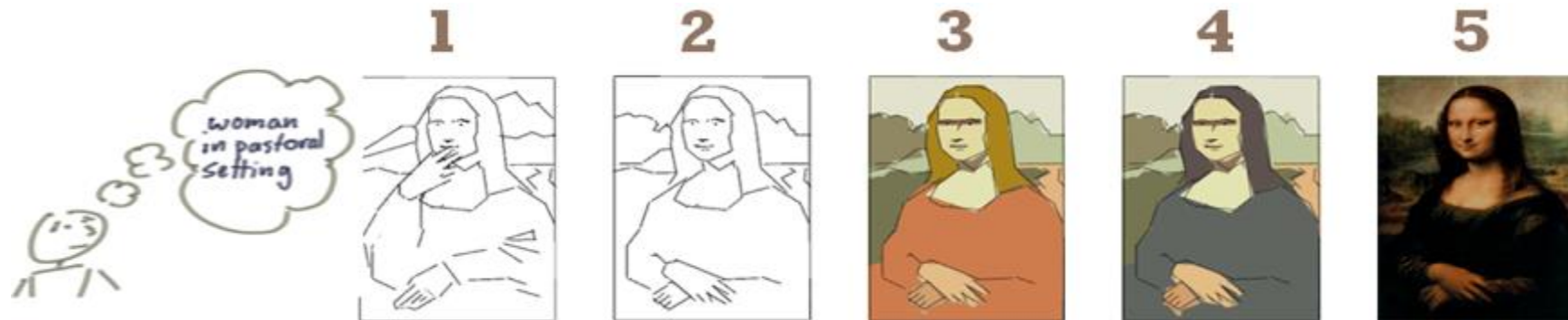


Rritëse vs Përsëritës (Incremental vs Iterative)

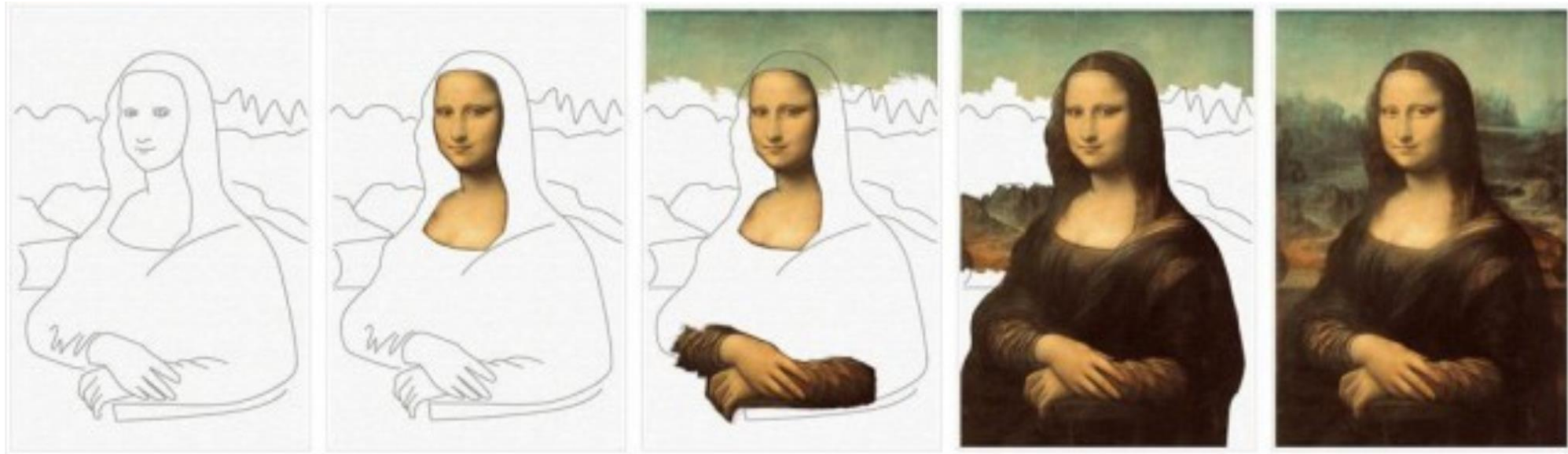
Incremental



Iterative



Rritëse & Përsëritës (Incremental & Iterative)...

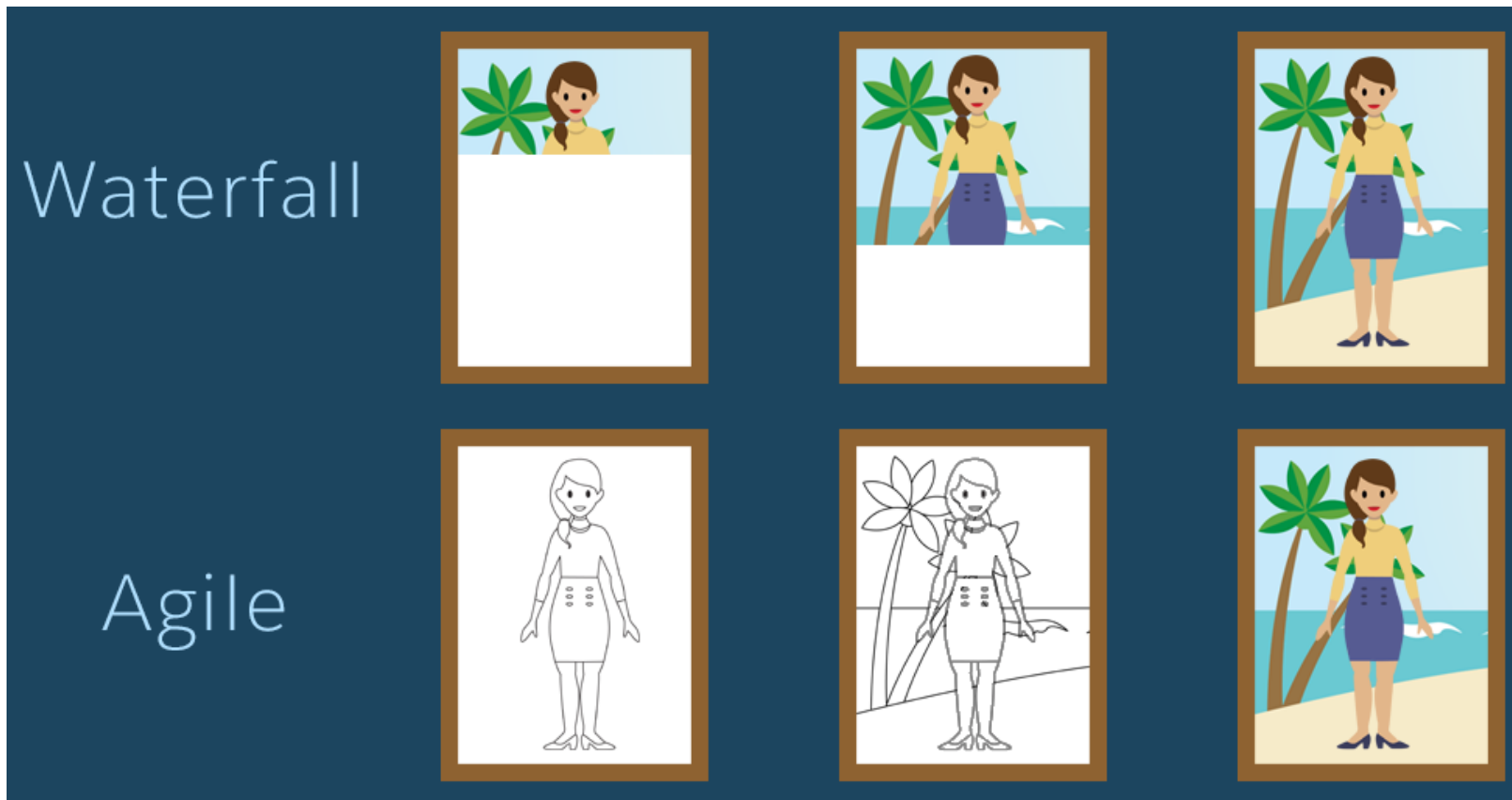


Agile = Iterative + Incremental

Don't try to get it all right
from the beginning

Don't build it all at once

An Iterative Waterfall Isn't Agile



Metodologjitë e zhvillimit të softuerit

Metodologjitë e zhvillimit të softuerit

□ Metodologjitë e zhvillimit të softuerit:

- Ad-hoc
- Waterfall / Ujëvarës
- Prototyping / Prototipit
- V-Model
- Spiral
- RAD
- Unified Process / Procesi i Unifikuar
- Agile/i Shkathët

Metodologjia Ad-hoc (Big Bang)

□ Në mënyrë tipike, "**nuk konsiderohet**" një lloj i metodologjisë!

- Klienti shpreh nevojat e tij, dhe zhvilluesi thjesht bën punën!

□ Përparsi:

- E thjeshtë dhe e drejtpërdrejtë
- E përshtatshme për **sisteme të vogla** dhe **të thjeshta**



□ Mangësi: Një problem i madh nëse:

- Klienti nuk i shpreh **saktë nevojat** e tij.
- Zhvilluesi **nuk di** si të **procedojë** me **implementimin**
- ... dhe kjo **nuk ka mbështetjen** e ndonjë **ndryshimi** në **kërkesat**

Metodologjia Waterfall / Ujëvarës

□ Model linear

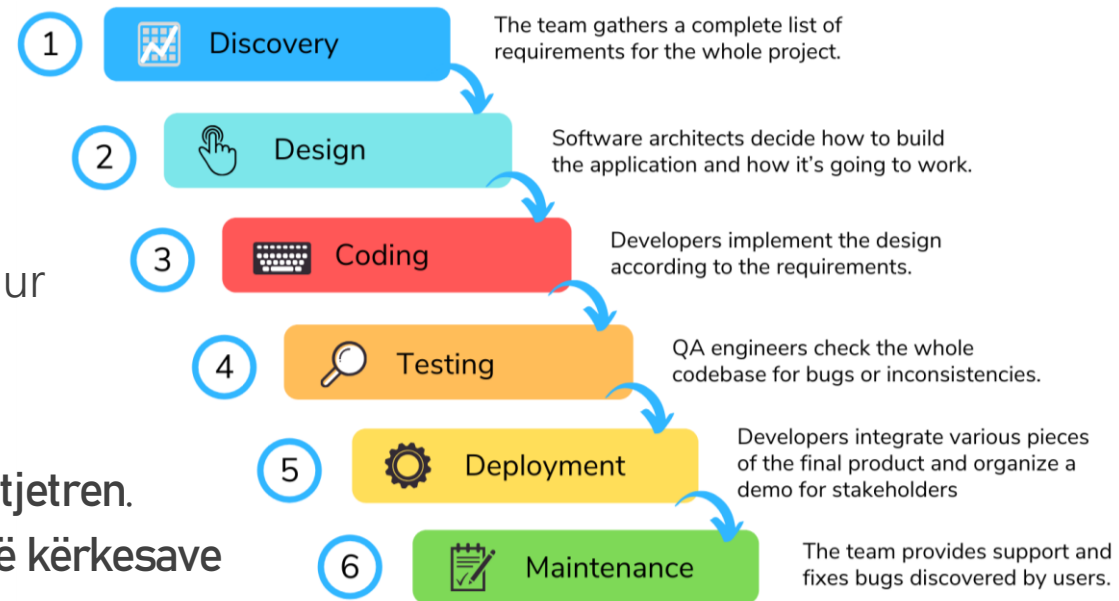
- Testimi bëhet pasi kodi të zhvillohet plotësisht

□ Përparsi

- Kërkesa është e qartë para se të fillojë zhvillimi (kodimi)
- Secila fazë përfundon në një periudhë specifike kohore.
- Si model linear, është i lehtë për tu zbatuar
- Secila fazë rezulton me një dokumentim përkatës të duhur

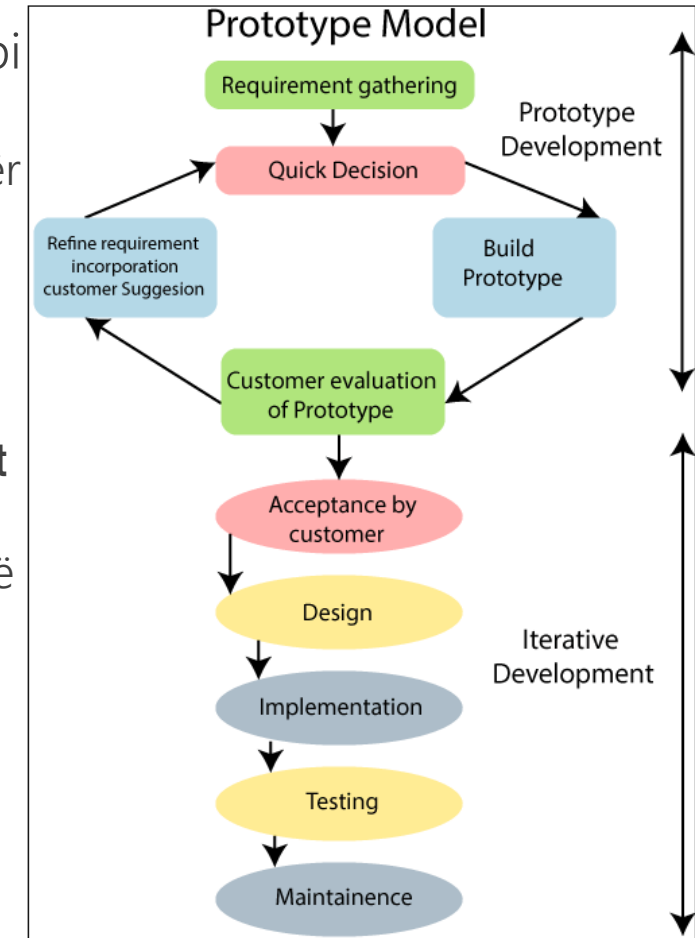
□ Mangësi:

- Mund të shfaqen probleme kur kalojmë nga një faza në tjetren.
- Nuk është fleksibil dhe nuk mbështet ndonjë ndryshim të kërkesave
- Klienti e sheh produktin pasi të mbarojnë të gjitha fazat.



Metodologjia e Prototipit

- ❑ **Metodologjia e Prototipit** ofron një version inicial të një prototipi të aplikacionit
 - **Simulon** vetëm *disa aspekte* të **produktit përfundimtar** dhe mundesi për të *identifikuar* problemet dhe zgjidhjet e mundshme.
- ❑ **Përparsi**
 - **Përfshirje** më e madhe e **përdoruesve/klienteve** në *produkt*, para implementimit të tij.
 - **Përdoruesit/klientet** e kanë më të qartë se **çfare do të ofroj softuerit** që po zhvillohet.
 - Ulet **koha** dhe **kostot** pasi që **defektet** mund të identifikohen shumë më herët.
 - **Feedback** më të shpejta nga **përdoruesit** që rezultojnë në zgjidhje më të mira.
 - Funksionet që **mungojnë** ose ato **konfuze** apo të **vështira** mund të identifikohen lehtësisht.
 - Ide të reja për dizajnim/funksione.



Metodologjia e Prototipit...

❑ Mangësi:

- Rreziku i analizës të pamjaftueshme së **kërkesave** për shkak të *varësisë* shumë nga prototipi.
- Përdoruesit mund të **ngatërrohen** me prototipet dhe sistemet reale.
- Praktikishtë, kjo metodologji mund të **rrisë kompleksitetin** e sistemit pasi **qëllimi** i sistemit mund të zgjerohet përtej planeve **origjinale**.
- Zhvilluesit mund të përpiqen të **ripërdorin** prototipet **ekzistuese për të ndërtuar sistemin e vërtetë**, edhe kur nuk është teknikisht i realizueshëm.
- **Investimet** në prototip mund të jetë *larta* nëse nuk kontrollohet si duhet.

Metodologjia e Modelit-V

❑ Versioni i **modifikuar** i metodës së ujëvarë (modeli linear)

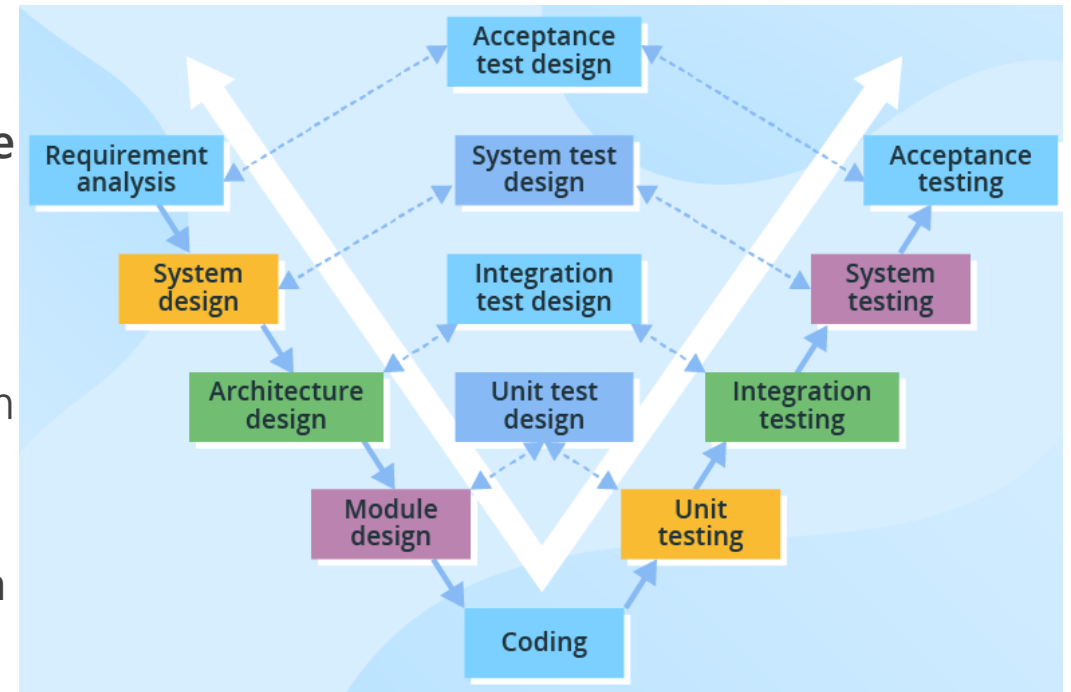
- Fazat janë *pasqyruar* për të **verifikuar** dhe **validuar** secilen prej tyre.
- **Testuesit** dhe **zhvilluesit** *punojnë paralelisht*, **test Rastet** përgatiten për **secilen fazë**.

❑ **Përparsi**

- Njëjt si **ujëvara** + **testuesi** është i përfshirë nga **faza e kërkesës**.

❑ **Mangësi:**

- I ngurtë
- Nëse **ndryshimet** ndodhin në mes të rrugës, jo vetëm **dokumentacioni i kërkesave**, por edhe **dokumentacioni i testimit** duhet të ndryshohet
- **Nuk** është i **përshtatshëm** për **projekte afatshkurtra** pasi kërkon *rishikime në secilën fazë*



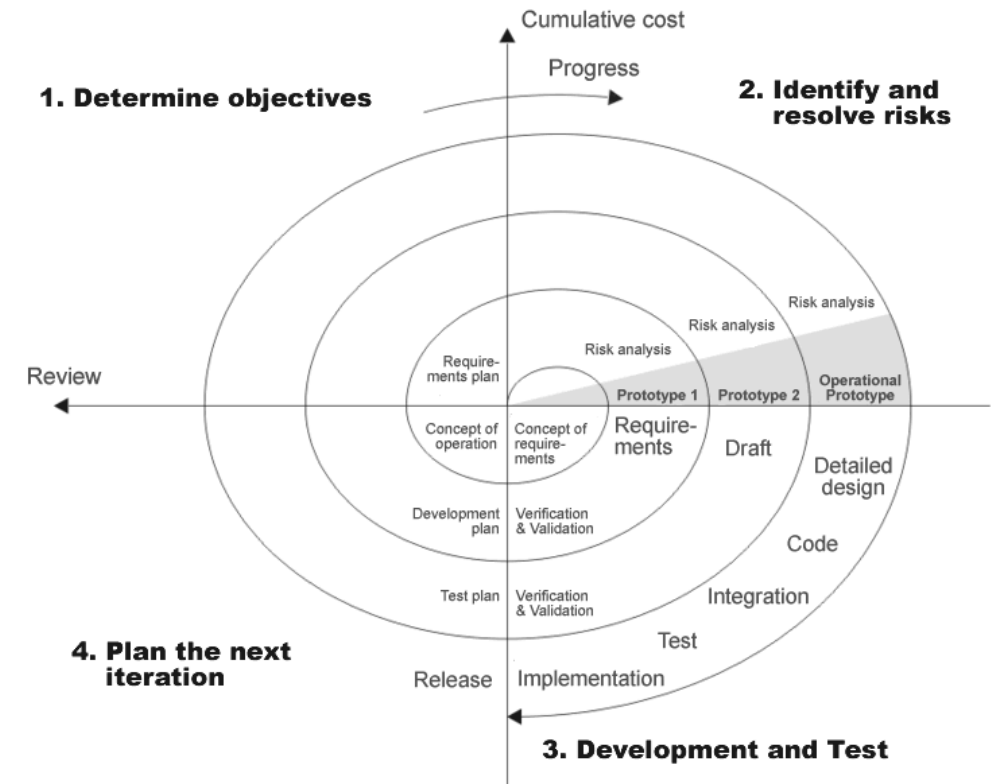
Metodologjia Spiral-e

❑ Modeli linear dhe rritës + Prototipi

- Focus në vlerësimin e rrezikut
- 4 faza kryesore:
 - Percaktimin e Objektivave, Analizon Rreziket, Inxhinieria (*dizajn, kodim, testim*) dhe Planifikimi.
- Modeli spiral është i favorizuar për projekte të mëdha, të shtrenjta dhe të ndërlikuara

❑ Kur të përdorim modelin spiral:

- Kur vlersohet kostoja dhe rreziku është i rëndësishëm.
- Kur krijimi i një prototipi është i përshtatshëm
- Kërkesat janë komplekse
- Projektet konsiderohen në rrezik nga niveli mesëm në të lartë
- Priten ndryshime të rëndësishme/të mëdha
- Përdoruesit janë të pasigurt për nevojat e tyre



Metodologjia Spiral-e...

❑ Përparsi:

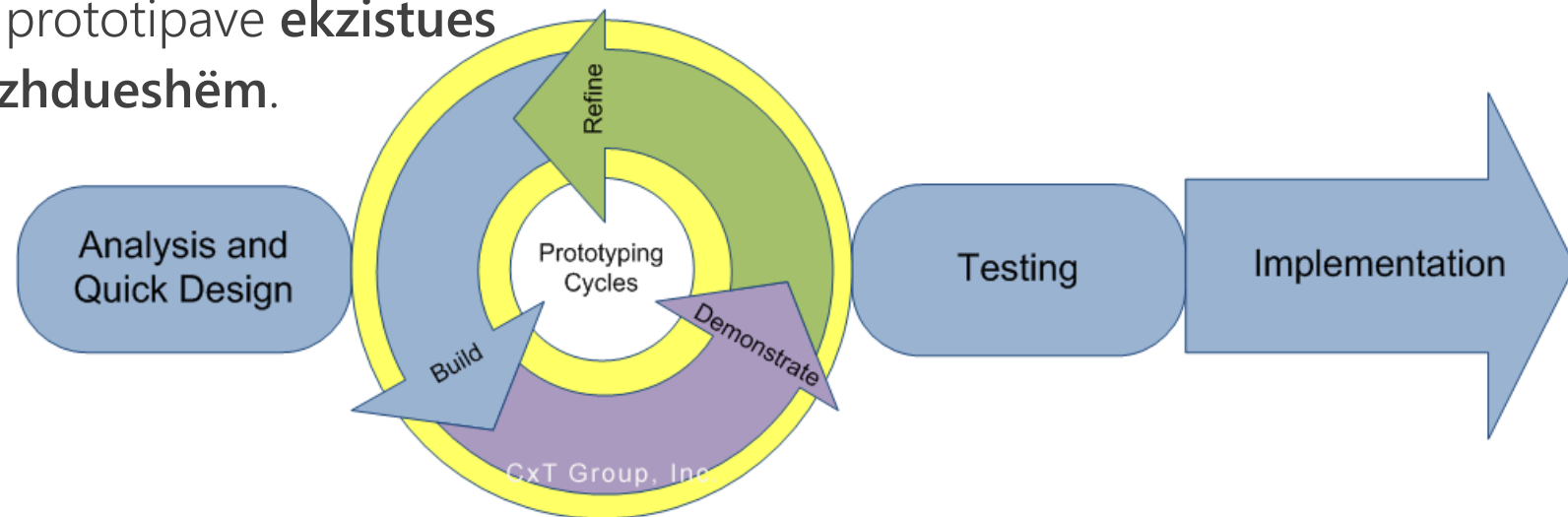
- Sasi të lartë të analizës së rrezikut.
- **Feedback** të *hershme* dhe të shpeshta nga përdoruesit si rezultati i përdorimeve të prototipeve
- Softueri ofrohet në **fazat e hershme**

❑ Mangësi:

- Mund të jetë i **kushtueshëm**
- Analiza e rrezikut kërkon **ekspertizë** shumë **specifike**
- **Nuk është adekuat** për projekte të **vogla**

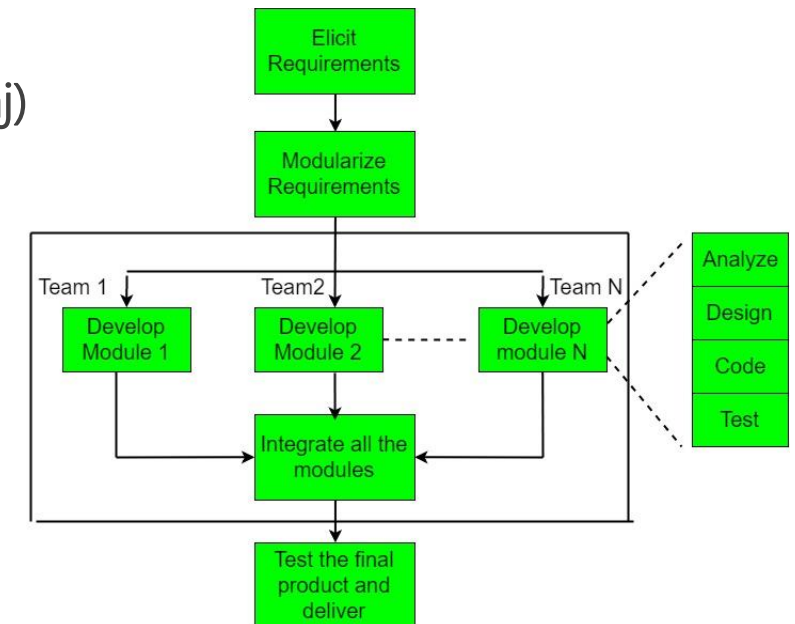
Metodologjia RAD: Rapid Application Development

- ❑ RAD: Zhvillimi i shpejtë i aplikacionev (Iterative, incremental dhe adaptative)
 - Përdor **planifikim minimal** në favor të **prototipeve** të shpejta.
- ❑ Mbledh kërkesat e klientëve përmes:
 - **Workshops** ose **fokus grupeve**
 - **Testimi i hershëm** i *prototipeve* duke përdorur konceptin **përsëritës**
 - **Ripërdorimi** i prototipave **ekzistues**
 - **Integrimi i vazhdueshëm**.



Metodologjia RAD: Rapid Application Development...

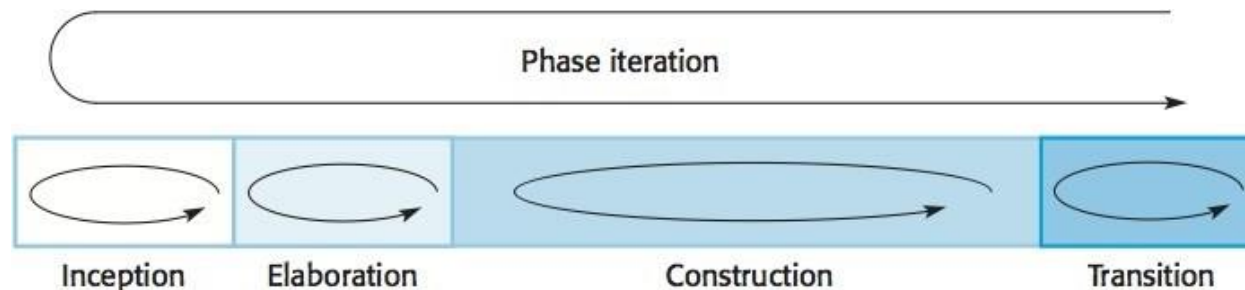
- ❑ Projekti mund të ndahet në module të vogla ku çdo modul mund të caktohet në mënyrë të pavarur ekipeve të veçanta.
- ❑ Kur përdoret Metodologjia RAD?
 - Kur një sistem duhet të prodhohet në një kohe të shkurtër (2-3 muaj)
 - Kur dihen kërkesat
 - Kur përdoruesi do të përfshihet gjatë gjithë ciklit jetësor të projektit
 - Kur rreziku teknik është më i vogël
 - E zbatueshme për sistemet modulare
 - Kur një buxhet është mjaft i lartë për të përballuar dizajnuesit për modelim së bashku me koston e mjeteve të automatizuara për gjenerimin e kodit.



Metodologjia UP (Procesi i Unifikuar)

□ PU – model Iterative dhe incremental, i drejtuar në USE CASE dhe i orientuar ne arkitekturen e sistemit.

- 1) **Inception (Zanafilla)**, ka për qëllim të vendos **këkesat e biznesit** për sistemin. Gjatë kësaj faze **identifikohen entitetet e jashtme** (njerëzit dhe sistemet) që do të ndër veprojnë me sistemin që po e ndërtojmë
- 2) **Elaboration (Elaburimi)**, ka për qëllim të **kuptuarit e problemit**, **krijimin konceptual të arkitekturës** së sistemit dhe krijimin e **planit të zhvillimit**.
- 3) **Construction (Ndërtimi)**, është fazë ku **dizajnohet**, **programohet** dhe **testohet** sistemi
- 4) **Transition (Trazicioni)** është faza finale e RUP ku produkti **kalon** në **shfrytëzim** (deployed)



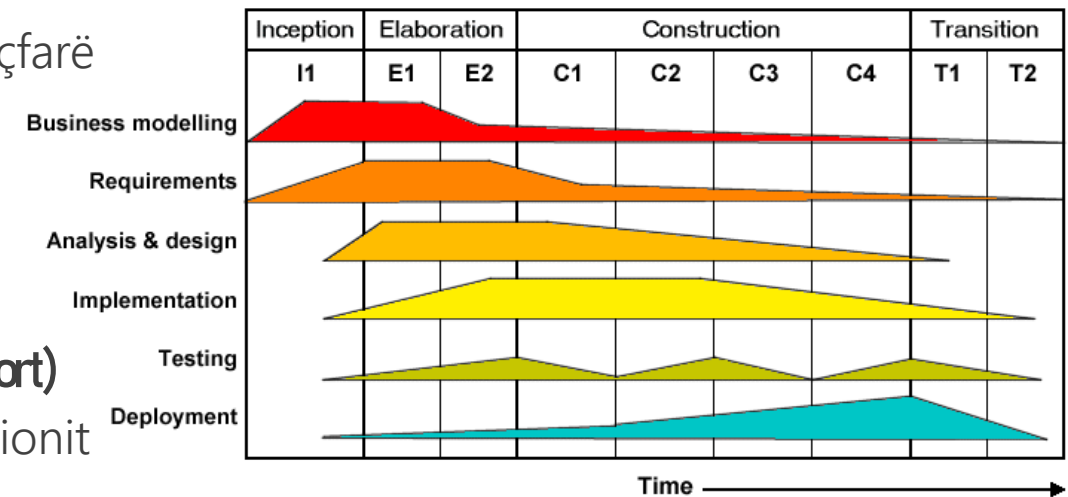
Metodologjia UP (Procesi i Unifikuar)

□ Inception (Zanafilla): 10% kohës, 5% përpjekje (Effort).

- Ndhmon **për** të përcaktuar fizibilitetin e projektit dhe çfarë dëshiron konsumatori.
- **Identifikimi e funksioneve kryesore** të sistemit
 - Një modelim fillestar të Use Case's (10-20% të sistemit)
 - Një ose disa prototipa.

□ Elaboration (Elaburimi) : 30% kohës, 20% përpjekje (Effort)

- Definon dhe përcakton arkitekturen në bazë të aplikacionit
- I kuptojm më thell kërkesat
 - Modelim të plotë të 80% Use Casave.
 - Përcaktojm kërkesa Jo-Funksionale
- Përshkrimi i arkitekturës së softuerit
- Prototipi arkitektonik i ekzekutueshëm



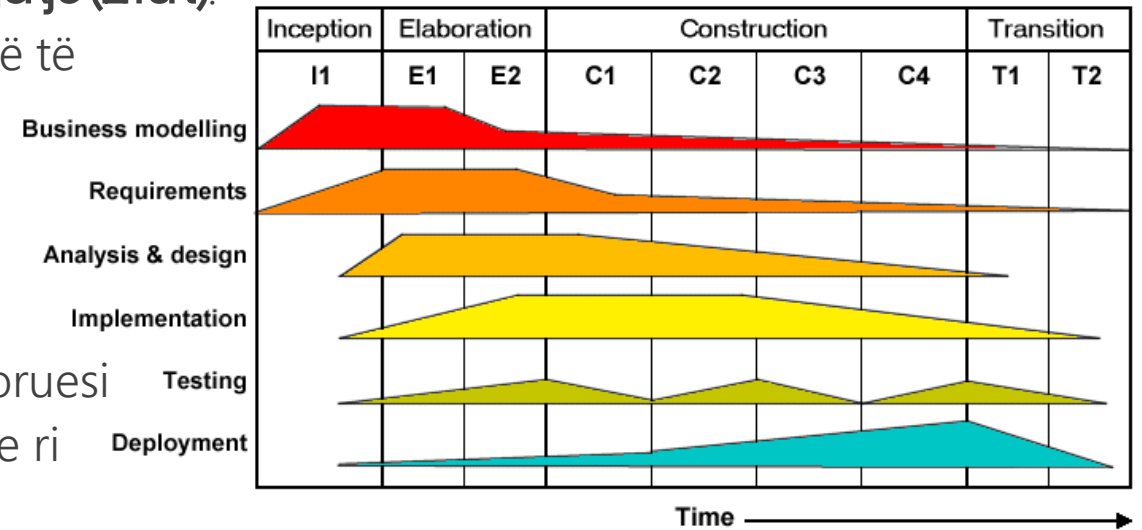
Metodologjia UP (Procesi i Unifikuar)...

□ Construction (Ndërtimi): 50% kohës, 65% përpjekje (Effort).

- Iterativisht kompletin zhvillimin/kodimin e plotë të produktit.
- Integron aplikacionin në platforma adekuate
- **Ofron** manuale të përdorimit.

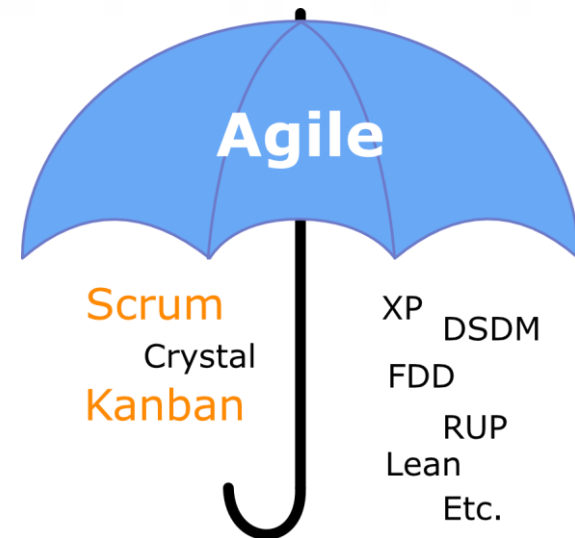
□ Transition: 10% kohës, 30% përpjekje (Effort).

- Dorzon produktin (aplikacionin) te klienti/përdoruesi
- Realizon testimin **Beta** për të vërtetuar sistemin e ri kundrejt përvojës së përdoruesit
- Trajnimi i përdoruesve dhe mirëmbajtësve



Metodologjia Agile

- ❑ Grupi i metodave të zhvillimit të softverit
- ❑ Bazuar në zhvillimin përsëritës+ritje
- ❑ Fraza më të rëndësishme:
 - **ekipe vetë-organizuese, ndër-funksionale**
 - **planifikim adaptues,**
 - **zhvillimi evolucionar dhe dorzim (delivery),**
 - Një qasje **përsëritëse (iterative)**, bllok-kohe (time-boxed), e orientuar drejt njerëzve dhe e përqendruar në rezultat,
 - **dorzimi i softverit** në mënyrë graduale
 - **reagim (feedback)** i shpejtë dhe **fleksibël** ndaj ndryshimeve.
- ❑ Një kornizë konceptuale



Metodologjia Agile...

- Metodologjia e **Agile** thekson **4 vlera** dhe **12 parime** për zhvillimin e softuerit.

THE AGILE MANIFESTO

Individuals and Interactions

over processes and tools

Working Software

over comprehensive documentation

Customer Collaboration

over contract negotiation

Responding to Change

over following a plan

Metodologjia Agile...

Nga 4 vlerat e Agile rrjedhin 12 parimet e mëposhtme

1, Our highest priority is to satisfy the customer through **early and continuous delivery** of valuable software.

2,, Welcome changing requirements **even late in development**.
Agile processes **harness change** for the customer's competitive advantage.

3, **Deliver working software frequently**, from a couple of weeks to a couple of months, with a **preference to the shorter timescale**.

4, **Business people and developers** must work together **daily** throughout the project.

5, Build projects around motivated individuals. **Give them the environment** and support they need, and **trust them to get the job done**.

6, The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.

7, **Working software** is the primary measure of progress.

8, Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9, Continuous attention to technical excellence and **good design enhances agility**.

10, Simplicity - the art of **maximizing the amount of work not done is essential**.

11, The best architectures, requirements, and designs emerge from **self-organizing teams**.

12, At regular intervals, the team **reflects on how to become more effective**, then tunes and adjusts its behaviour accordingly.

Metodologjia Agile...

□ Benefitet:

- **Fokusim ne Perdorues** – Tregim të Perdoruesit me fokus biznesor duke determinuar kriteret e pranimi për karakteristikat/funksionet e produktit/aplikacionit.
- **Fokusim në vlerën e Biznesit**– dorzimi i pjese së aplikacioni (funksioneve) që ofron vlerë biznesore bazuar në rëndesine e biznesit të klientit.
- **Përmirëson Cilësinë** - fokusohet në cilësi të lartë të zhvillimit, testimit dhe bashkëpunimit duke e ndarë projektin në njësi të menaxhueshme.
- **Transparenca** - mundësi që klientët të përfshihet dhe të monitorojnë gjatë gjithë proceseve në projekt.
- **Dorëzimi i hershëm dhe i parashikueshëm** - shërbime, produkte ose funksione të reja të cilat dorëzohen ne orar fiks sipas Sprintave.
- **Parashikim të kostos dhe orarit**- orar-fiks, klienti mund të kuptojë koston e përafërt të secilit funksion, prandaj përmirëson vendimmarrjen, jep përparësi funksioneve dhe rendesin për iteracione shtesë.
- **Lejon ndryshimin** -Ekipet bëjnë ndryshime në mënyrë që të përmirësojnë funksionet dhe efektivitetin. Ndryshimet mund të alokohen ne iteracionin e ardhshëm.
- **Inkuadrim i palëve të interesuara**. Palët e interesuara dhe zhvilluesit punojnë ngushtë çdo ditë.

Faleminderit...!

